1.0

2.8    2.5
3.2    2.2
3.6

1.1              2.0

1.8

1.25    1.4    1.6

NPS-53-81-002

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

SMOOTH INTERPOLATION OF SCATTERED DATA BY

LOCAL THIN PLATE SPLINES

Richard Franke

Technical Report for Period Jan 79 - Mar 81

81 4 27 043

NAVAL POSTGRADUATE SCHOOL
Monterey, California
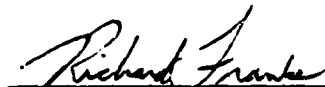
Rear Admiral Tyler F. Dedman
Superintendent

David A. Schrady
Acting Provost

Reproduction of all or part of this report is authorized.

This report was prepared by:


RICHARD FRANKE
Associate Professor
Department of Mathematics


Reviewed by:                      Released by:


CARROLL O. WILDE, Chairman        W. M. TOLLES
Department of Mathematics         Dean of Research

| REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|

| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
|---|---|---|
| NPS-53-81-002 | AD-A098232 | |

| 4. TITLE (and Subtitle) | 5. TYPE OF REPORT & PERIOD COVERED |
|---|---|
| Smooth Interpolation of Scattered Data by Local Thin Plate Splines. | Technical Report - Jan 1979 — Mar 1981. |
| | 6. PERFORMING ORG. REPORT NUMBER |

| 7. AUTHOR(s) | 8. CONTRACT OR GRANT NUMBER(s) |
|---|---|
| Richard Franke | |

| 9. PERFORMING ORGANIZATION NAME AND ADDRESS | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
|---|---|
| Foundation Research Program Naval Postgraduate School Monterey, CA 93940 | |

| 11. CONTROLLING OFFICE NAME AND ADDRESS | 12. REPORT DATE |
|---|---|
| Chief of Naval Research Arlington, VA 22217 | 26 March 1981 |
| | 13. NUMBER OF PAGES |
| | 30 |

| 14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office) | 15. SECURITY CLASS. (of this report) |
|---|---|
| | Unclassified |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Scattered data interpolation
Interpolation
Smooth interpolation
Bivariate interpolation

Multivariate approximation
Thin Plate Splines

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An algorithm and the corresponding computer program for solution of the scattered data interpolation problem is described. Given points $(x_k, y_k, f_k)$, $k = 1, \ldots, N$ a locally defined function $F(x,y)$ which has the property $F(x_k, y_k) = f_k$, $k = 1, \ldots, N$ is constructed. The algorithm is based on a weighted sum of locally defined thin plate splines, and yields an interpolation function which is differentiable. The program is available from the author.

## 1.0 Introduction

The problem of constructing interpolation functions for *sets of* scattered data in two independent variables has been treated in many papers. The survey paper on approximations to multivariate functions by Schumaker [20] contains extensive references. Other survey papers are by Barnhill [3] and Sabin [19]. The present author has surveyed and tested a number of algorithms for solution of the problem [10], [11]. Conceptually the problem is quite simple: Given points $(x_k, y_k, f_k)$, $k = 1, \ldots, N$, with distinct $(x_k, y_k)$, construct a function $F(x,y)$ so that $F(x_k, y_k) = f_k$, $k = 1, \ldots, N$. Generally one wants to have a smooth interpolant, $F(x,y)$, in the sense that low order partial derivatives are everywhere continuous. This is complicated *for* large sets of data by the fact that the interpolant (in a practical sense, to be computable) must be local, so that its value at some point $(x,y)$ depends only on $(x_k, y_k, f_k)$ values for which $(x_k, y_k)$ is "close" to $(x,y)$. A general framework for a class of such methods is given in [9], and we will discuss it briefly in Section 2.

While a large number of ideas have been proposed for solution of the problem, a much smaller number of working computer programs are readily available. These include: (1) a method based on finite element functions defined over triangles, due to Akima [1], [2], a version of which is available in the IMSL[*] library under the name IQHSCV, (2) a program based on a similar idea, due to Lawson [14], (3) two programs based on weighted local approximations by quadratic functions, due to Franke and Nielson [12]. A program by Little [15], and another by Nielson [18], both based on finite

1

element functions defined over triangles, will probably be available by press time. All of these programs have been tested by the author [10], [11], and most perform adequately in a variety of cases; None of them seems to have a clear edge over all the others, or to be entirely satisfactory. For certain applications, each has its good points. The choice of a method for most users will be based on subjective criteria which vary from person to person and application to application. It is not surprising this is the case in two variables since it is also the case in one variable, although perhaps to a lesser extent.

The purpose of this paper is to document an alternative scheme which performed comparably well in the previously mentioned tests. In this way the computer program will be brought to the attention of potential users, who may request it from the author. The method will supplement the currently available codes in that it is based on a different approach. It is anticipated that it will find application and approval in a variety of areas.

The theoretical background for the method is discussed in Section 2, while details of the program are outlined in Section 3. Section 4 gives information concerning usage of the program. Several examples are given in Section 5, including perspective plots of some surfaces generated by the program. Included is an example of how the variation of a parameter in the method affects the surface. General guidelines for choice of this parameter are given, even though the suggested value usually leads to satisfactory results. A general discussion of the features of the method is given, along with general guidance for use of the program.

## 2.0  Theoretical Background

The general idea encompassing this scheme and many others are given in [9]. Consider that the points $(x_k, y_k, f_k)$, $k = 1, \ldots, N$ are given. Briefly, local approximations to the data are constructed, and these are

2

then blended together by using weighting functions which form a partition of unity on $R^2$. We pause to give the necessary notation and definitions.

A set of functions, $w_i(x,y)$, $i = 1, \ldots, m$ is said to form a partition of unity if each $w_i(x,y) \geq 0$ and $\sum_{i=1}^{m} w_i(x,y) \equiv 1$. The $w_i$ will be called weight functions. Let the support of $w_i$ be denoted by $S_i =$ closure $\{(x,y): w_i(x,y) \neq 0\}$. Let $I_i = \{k: (x_k, y_k) \in$ Interior $(S_i)\}$. Now suppose that the functions $Q_i(x,y)$, $i = 1, \ldots, m$, are defined on $S_i$ and have the property that they interpolate the data whose $(x,y)$ coordinates are in Interior $(S_i)$, i.e., if $k \in I_i$, then $Q_i(x_k, y_k) = f_k$. These functions $Q_i$ will be called local approximations. We then consider the function $F(x,y) = \sum_{i=1}^{m} w_i(x,y) Q_i(x,y)$. Its properties are summarized in the following.

<u>Proposition.</u> The function $F(x,y) = \sum_{i=1}^{m} w_i(x,y) Q_i(x,y)$ has the following properties:

(1) Interpolation; $F(x_k, y_k) = f_k$, $k = 1, \ldots, N$.

(2) Smoothness; $F(x,y)$ is at least as smooth as the $w_i$ and $Q_i$, e.g., if all of the functions $w_i$, $Q_i$, $i = 1, \ldots, m$ have continuous first derivatives, so does $F(x,y)$.

(3) Local dependence on the data; Let $(x,y)$ be fixed, and let $J_{x,y} = \{i: w_i(x,y) \neq 0\}$, then $F(x,y)$ depends only on the $(x_k, y_k, f_k)$ points for which $k \in ( \cup_{i \in J_{x,y}} I_i) \cup \{i:$ some $Q_j$, $j \in J_{x,y}$ depends on $(x_i, y_i, f_i)\}$. In particular, we have $F(x,y) = \sum_{i \in J_{x,y}} W_i(x,y) Q_i(x,y)$.

These properties are essentially observations, but form the basis for construction of appropriate weight functions which will allow easy determination of the set $J_{x,y}$. Our construction yields a set of at most four nonzero terms in the sum defining $F(x,y)$. This provides a considerably faster process during the evaluation of the interpolant than was possible in the choices previously considered [9].

It has beer implied, but is not necessary for the proposition to

hold, that many of the weight functions, $w_i$, have finite support.  In order

for the local approximations $Q_i(x,y)$ to actually be local, this will

likely be the case.  Therefore we think of weight functions whose support,

$S_i$, contains relatively few $(x_k,y_k)$ points, and whose support is often

local.  In order for the interpolant to be defined on all of $R^2$, some sets

$S_i$ must not be finite, and typically would contain points $(x_k,y_k)$ on or

near the boundary of the convex hull of the set of points $\{(x_k,y_k)\}$.  With

this framework and ideas in mind, we are ready to discuss the specific

details of the algorithm.

## 3.0  Details of the Algorithm

### 3.1  Choice of Weight Functions

While the choice of weight functions was allowed to determine the

support regions in the discussion of the previous section, it is more

convenient to proceed from support regions to weight functions in the

actual application.  The use of support regions which are rectangles have

specific advantages in terms of controlling the number of support regions in

which a particular point $(x,y)$ lies, as well as simplifying determination

of those regions.

Let $n_x$ and $n_y$ be given positive integers, and let finite values of

$\tilde{x}_0 < \tilde{x}_1 < \tilde{x}_2 < \ldots < \tilde{x}_{n_x} < \tilde{x}_{n_x+1}$ and $\tilde{y}_0 < \tilde{y}_1 < \tilde{y}_2 < \ldots < \tilde{y}_{n_y} < \tilde{y}_{n_y+1}$ be

given.  For each $i = 1, 2, \ldots, n_x$ and $j = 1, 2, \ldots, n_y$ let $r_{i,j} = $

$[\tilde{x}_{i-1}, \tilde{x}_{i+1}] \times [\tilde{y}_{j-1}, \tilde{y}_{j+1}]$.

Let $H_3(s) = 1 - 3s^2 + 2s^3$, the Hermite cubic satisfying $H_3(0) = 1$,

$H_3(1) = H_3'(0) = H_3'(1) = 0$.  We then define piecewise cubics with continuous

4

first derivatives, which are nonzero only on two adjacent intervals, and
satisfy

$$v_i(x_j) = \delta_{ij} \quad , \quad i,j = 1, 2, \ldots, n_x$$

$$u_j(y_i) = \delta_{ji} \quad i,j = 1, 2, \ldots, n_y.$$

In particular we take

$$v_1(x) = \begin{cases} 1 & , \quad x < \tilde{x}_1 \\ H_3\left(\dfrac{x - \tilde{x}_1}{\tilde{x}_2 - \tilde{x}_1}\right) & , \quad \tilde{x}_1 \leq x < \tilde{x}_2 \\ 0 & , \quad x \geq x_2 \end{cases}$$

$$v_i(x) = \begin{cases} 0 & , \quad x < \tilde{x}_{i-1} \\ 1 - v_{i-1}(x) & , \quad \tilde{x}_{i-1} \leq x < \tilde{x}_i \\ H_3\left(\dfrac{x - \tilde{x}_i}{\tilde{x}_{i+1} - \tilde{x}_i}\right) & , \quad \tilde{x}_i \leq x < \tilde{x}_{i+1} \\ 0 & , \quad x \geq \tilde{x}_{i+1} \end{cases}$$

for $i = 2, \ldots, n_x - 1$

$$v_{n_x}(x) = \begin{cases} 0 & , \quad x < \tilde{x}_{n_x-1} \\ 1 - v_{n_x-1}(x) & , \quad \tilde{x}_{n_x-1} \leq x < \tilde{x}_{n_x} \\ 1 & , \quad x \geq \tilde{x}_{n_x} \end{cases}$$

The $u_i(y)$ are dual.  Then if we define

$$W_{i,j}(x,y) = v_i(x)u_i(y) \quad , \quad i = 1, \ldots, n_x \ , \quad j = 1, \ldots, n_y,$$

it is easily observed that $W_{i,j}$ has support $r_{i,j}$, except for when $i = 1$ or $n_x$, or $j = 1$ or $n_y$, when the support extends to $\infty$ in one or both variables. We denote the support of $W_{i,j}$ by $R_{i,j}$. Further, we note that the $W_{i,j}$ form a partition of unity on $R^2$.

Since the $\bar{x}_i$ and $\bar{y}_j$ values give rise to a grid on the plane, we call them grid values. The choice of grid values $\bar{x}_i$ and $\bar{y}_j$ depend on the data. They may be specified by the user, but the default option is for them to be determined by the program. In the default mode the user specifies a parameter, NPPR (for number of points per region), the suggested value being about 10. The program will then determine the grid values so that the anticipated number of $(x_k, y_k)$ points in each rectangle $R_{i,j}$ will be approximately NPPR. For data which is not somewhat uniformly distributed the actual numbers may vary considerably. However, for most situations we have encountered, the process is quite adequate.

An equal number of grid values is taken in each direction, i.e., $n_x = n_y$. Because we want NPPR points per rectangle, each subrectangle $(x_i, x_{i+1}) \times (y_j, y_{j+1})$ should have $\frac{1}{4}$ NPPR points. Since $n_x = n_y$, we want $(n_x + 1)^2 \frac{1}{4}$ NPPR $= N$, leading us to take $n_x$ to be the nearest integer to $(4N/NPPR)^{1/2} - 1$.

The grid values $\bar{x}_i$, are now chosen so that there are approximately equal numbers of points from the values $x_k$, $k = 1, \ldots, N$ in each interval $(\bar{x}_i, \bar{x}_{i+1})$. Specifically, let $\hat{x}_k$ denote the $x_k$ values arranged in non-decreasing order. Consider the points $(0, \hat{x}_1)$, $(1, \hat{x}_2)$, $\ldots$, $(N-1, \hat{x}_N)$, and let $g(t)$ be the piecewise linear interpolant for them. Divide the interval $(0, N-1)$ into $n_x + 1$ subintervals of length $\Delta = \frac{N-1}{n_x + 1}$. The values of $\bar{x}_i$ are determined by taking them to be the values of $g$ at the endpoints of the subintervals, i.e., $\bar{x}_i = g(i)$, $i = 0, 1, 2, \ldots, n_x + 1$. The $\bar{y}_j$ are determined in dual fashion. This process results in the grid values and hence the rectangles, being symmetric if the $(x_k, y_k)$ points are symmetric. In addition, the relative positions of grid values are unchanged by linear displacements

6

and stretching in each variable.

When chosen in the above fashion, the location of the lines is not
a local process in the sense that insertion of an additional point will
change the boundaries of all of the rectangles. While one could argue that
then the scheme is not local, we take the view that the idea of local
determination of the interpolant is most important in the evaluation phase.
The determination of parameters in the scheme (here, the $\bar{x}_i$ and $\bar{y}_j$) may be
a global process. Of course, if the user specifies the grid values, he will
likely be using a global process to choose them.

## 3.2  Choice of Local Approximations

The only constraint on the local approximations is that they interpolate
the appropriate points, and that they have continuous first derivatives
(at least) to assure a smooth interpolant. In the previously mentioned tests
conducted by the author, a number of global interpolation schemes for scattered
data were considered. In principle, any of these might be used. The choice
here was made for two reasons, (1) the method scored very well in the tests,
and (2) the method has an elegant and well developed mathematical theory which
also has direct application to some engineering problems.

The local approximations used in this algorithm are the thin plate splines
first mentioned by Harder and Desmairis [13], with theoretical developments
by Duchon [4-7] and Meinguet [16], [17]. It was first developed as the
solution to the problem of a thin plate which is forced to pass through
certain points (the interpolation points) by application of point loads. For
our purposes it is sufficient to know that the approximation is of the form

$$Q(x,y) = \sum_{k \in I} A_k d_k^2 \log d_k + a + bx + cy ,$$

where $I = \{k:$ Q is to take on the value $f_k$ at $(x_k, y_k)\}$, and $d_k^2 = (x - x_k)^2 + (y - y_k)^2$. The coefficients $A_k$, and a, b, and c are determined by the linear

7

system of equations

$$\sum_{k \in I} A_k d_k^2 \log d_k + a + bx + cy \Bigg|_{(x,y) = (x_i, y_i)} = f_i , \qquad i \in I$$

$$\sum_{k \in I} A_k = 0$$

$$\sum_{k \in I} A_k x_k = 0$$

$$\sum_{k \in I} A_k y_k = 0$$

The geometric effect of the last three equations is to suppress all terms in the approximation which grow faster than linear as distance from the interpolation points is increased. A linear system of order equal to the number of interpolation points plus three must be solved. To be nonsingular there must be at least three noncollinear points among the $(x_k, y_k)$, $k \in I$. The system is symmetric, but not positive definite. While an equation solver designed for such systems could be used, we have found a general purpose solver, the DECOMP/SOLVE subroutines of Forsythe, Malcolm, and Moler [8] has given more reliable results.

It is easily observed that the local interpolant has continuous derivatives of all orders except at the data points, $(x_k, y_k)$, where a logarithmic singularity occurs in the second derivatives. The interpolant has linear precision, that is, if the $(x_k, y_k, f_k)$ points all lie on a linear function, the interpolant will reproduce it.

While the thin plate spline is invariant with respect to scaling, translation, and rotation (not all of this is obvious), the condition number of the coefficient matrix for the system of equations is dependent on scaling. To minimize difficulties with that, and to remove the effect of scaling the variables by different amounts, we transform each rectangle $r_{i,j}$ onto the unit square $[0,1]^2$, before the local approximation $Q_{i,j}$ is computed.

It remains to specify the process for determining the points $(x_{k_k}, z_k)$ to be interpolated by the thin plate spline local approximation. Experience has shown that it is advantageous to include more points than is necessary, i.e., $(x_k, y_k)$ which are outside of $R_{i,j}$. This tends to yield a better transition between local approximations than when only necessary points are included. The set of $(x_k, y_k)$ points transformed into the rectangle $R = [-.1125, 1.1125]^2$ by the transformation taking $r_{i,j}$ onto $[0,1]^2$ are included. Let

$$I_{i,j} = \left\{ k: \left( \frac{x_k - \tilde{x}_{i-1}}{\tilde{x}_{i+1} - \tilde{x}_{i-1}} \;,\; \frac{y_k - \tilde{y}_{j-1}}{\tilde{y}_{j+1} - \tilde{y}_{j-1}} \right) \varepsilon \, R \right\}.$$

This gives the basic set of interpolation points for the local approximation $Q_{i,j}$ associated with the rectangle $R_{i,j}$. Under certain conditions there may be fewer than the necessary three indices in $I_{i,j}$. When this happens, the set $I_{i,j}$ is augmented by including as interpolation points the necesary number of closest points to the rectangle $R_{i,j}$ ( in the $\ell_\infty$ norm), after the points $(x_k, y_k)$ have undergone the transformation to the unit square. The minimum number of points per rectangle is a variable, MINPTS. This has been set to 3, but may be increased if it seems desirable.

After the interpolation points for each local approximation have been determined, the local thin plate splines, $Q_{i,j}$, can be determined by calculating the coefficients. This yields

$$Q_{i,j}(x,y) = \sum_{k \varepsilon I_{i,j}} A_{i,j,k} \, d_k'^2 \log d_k' + a_{i,j} + b_{i,j} \, x' + c_{i,j} \, y'$$

where the primes denote coordinates and distance after the transformation of $r_{i,j}$ to the unit square

## 3.3 Properties of the Interpolant

The overall interpolant is of the form

$$F(x,y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} W_{i,j}(x,y) Q_{i,j}(x,y),$$

9

and as noted previously, there are at most four nonzero terms in the sum. Which terms are nonzero is easily determined. If $x \geq \bar{x}_{n_x}$, set $i' = n_x$; Otherwise let $i'$ be the smallest index so that $\bar{x}_{i'+1} > x$. Determine $j'$ in dual fashion from $y$ and the $\bar{y}_j$'s. Then, the four nonzero terms have indices $(i',j'),(i'+1,j'),(i',j'+1)$, and $(i'+1,j'+1)$. If $i' = 0$ or $i' = n_x$, the terms involving $i'$ or $i'+1$, respectively, do not appear, and similarly if $j' = 0$ or $j' = n_y$, the terms involving $j'$ or $j'+1$, respectively, do not appear.

In dition to the properties outlined in Section 2, certain other properties hold. The approximation is invariant under translation and stretching (independently in each variable). It has symmetry with respect to planes parallel to coordinate planes whenever the data has that symmetry. The approximation is not invariant under rotations, however, since the rectangles depend on the individual coordinates of the data points.

The approximation has continuous first derivatives, and jump discontinuities in the second derivatives across grid lines, as well as logarithmic singularities in the second derivatives at the data points. Plots of the surfaces generally appear to be quite smooth, however. Since the local approximations have linear precision, the overall approximation also has linear precision, i.e., if the data lies on a linear function, the interpolant is a linear function.

## 4.0  The Computer Program

The overall hierarchy of the subroutines is given in Diagram 1, which shows the communication links between them. No COMMON is used as array sizes are problem dependent and thus specified by the user. We briefly discuss them, mentioning important parameters.
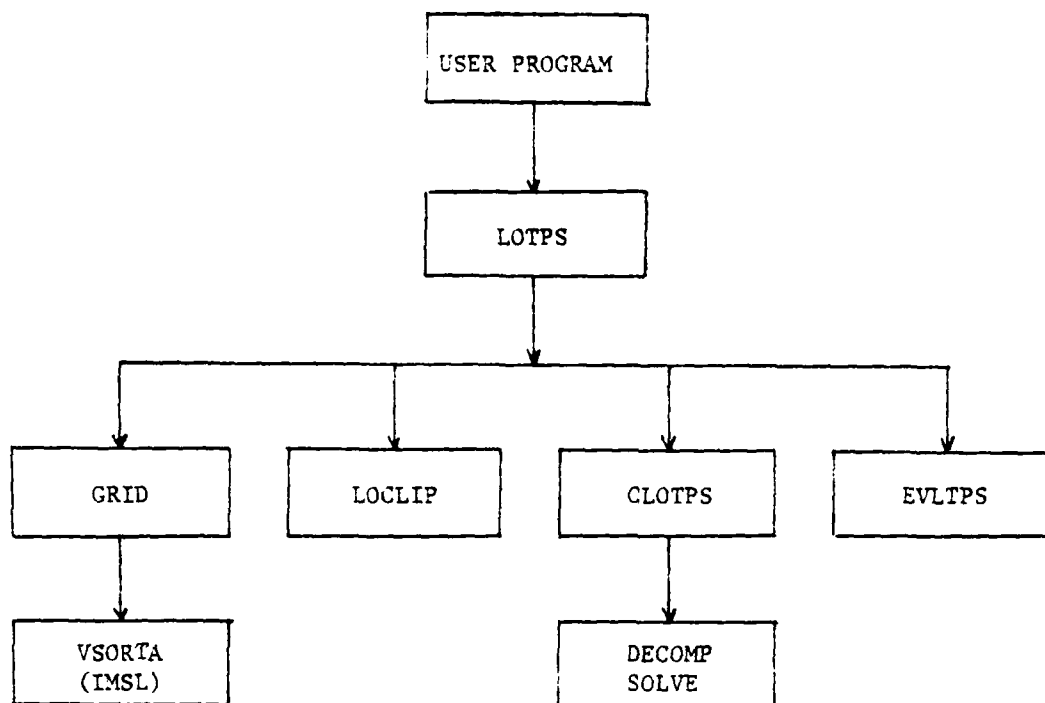
10

```
        ┌─────────────────┐
        │  USER PROGRAM   │
        └────────┬────────┘
                 │
                 ▼
        ┌─────────────────┐
        │     LOTPS       │
        └────────┬────────┘
                 │
    ┌────────┬───┴────┬────────┐
    ▼        ▼        ▼        ▼
┌───────┐┌───────┐┌───────┐┌───────┐
│ GRID  ││LOCLIP ││CLOTPS ││EVLTPS │
└───┬───┘└───────┘└───┬───┘└───────┘
    │                 │
    ▼                 ▼
┌───────┐         ┌───────┐
│VSORTA │         │DECOMP │
│(IMSL) │         │SOLVE  │
└───────┘         └───────┘
```

Diagram 1.

## Calling Program

This program is supplied by the user and must supply the $(x_k, y_k, f_k)$ points, plus two arrays of points $x0_i$, and $y0_j$ for the grid of points $(x0_i, y0_j)$ at which the interpolant is to be evaluated. In addition, the user must supply two workspace arrays, IWK and WK in which information calculated during preprocessing (e.g., $I_{i,j}$, $\tilde{x}_i$, $\tilde{y}_j$, and coefficients for the local approximations), is stored, and an array FO for the returned interpolant values.

The amount of storage required for the arrays IWK and WK is not known a priori. The estimated space required is about 6N for IWK and about 7N for WK. Table 1 gives exact results for several different sized problems based on random (x,y) points. Oddly distributed point sets may result in somewhat more storage being required. In any case, the precise

11

number of locations required is returned to the calling program from Subroutine LOTPS, the only routine referenced by the user. If an insufficient number are allowed, an error return occurs.

Under the usual option, the user specifies NPPR > 0, the suggested value being 10. If the user wishes to specify the grid lines, he may do so by setting NPPR = 0, and then giving grid line information in the arrays IWK and WK, as explained in the argument description for Subroutine LOTPS. Typically one should take $\bar{x}_0 = \min_k x_k$, $\bar{x}_{n_x+1} = \max_k x_k$, and the dual in y. This is not necessary, although all points to be interpolated should lie in $[\bar{x}_0, \bar{x}_{n_x+1}]$ $[\bar{y}_0, \bar{y}_{n_y+1}]$. To prevent different scaling (internally) in the two variables, a square grid covering the $(x_k, y_k)$ points could be specified.

### Subroutine LOTPS

This subroutine provides the interface between the user's program and the set of routines implementing the method. Generally, LOTPS sets up storage areas in the arrays IWK and WK, determines parameters required by other subroutines, and calls other subroutines to (1) generate the grid, if necessary, (2) determine the interpolation points for the local approximations, (3) compute coefficients for the local approximations, (4) evaluate the interpolant at a grid of points.

### Subroutine GRID

This subroutine selects values of $\bar{x}_i$ and $\bar{y}_j$ in accordance with the discussion in Section 3.

### Subroutine LOCLIP

This subroutine determines the interpolation points for each local interpolant, in accordance with Section 3.

### Subroutine CLOTPS

This subroutine generates the system of equations for the coefficients of the local approximations and calls an equation solver to obtain them.

12

Internally the routine has a maximum of 30 local interpolation points. This can easily be altered by changing two statements, as noted in the program listing.

## Subroutine EVLTPS

This subroutine evaluates the interpolant on the grid of points specified by the user. Use of a grid, when that is required, facilitates the process of locating the rectangle in which a particular evaluation point is located. The $x0_i$ and $y0_i$ values should be in increasing order for maximum efficiency. Evaluation time for a grid of points should be nearly independent of N for large N, which is borne out by the timing information in Table 2.

## 5.0   Examples and Observations

Example 1. This example shows a typical local approximation function and is for the data in Table 1. This function is a "cardinal" function for the first point, and as such shows the effect of a nonzero value at a single point on the interpolant. The plotted surface, shown in Figure 1, is over $[0,1]^2$.

Example 2. This example is given to show a surface generated from 100 randomly generated points with the function value being obtained from an explicitly given function. The surface was generated using NPPR = 10, has rms error of about .3%, and is virtually indistinguishable from a plot of the parent surface. It is shown in Figure 2.

Examples 3, 4, 5. These examples use the same set of 60 points lying in the square $( -\frac{1}{18} , \frac{19}{18} )^2$, and chosen by a pseudorandom number generator. The function is explicitly given by $f(x,y) = .1 + \frac{\sin 3(x+y)}{12(x+y)}$ , and is shown in Figure 3. Figures 4, 5, and 6 show the interpolant over the square $[0,1]^2$ for NPPR values of 6, 10, and 15.

From these examples we see that NPPR = 10 works well, not too much difference is observed when NPPR is increased, but NPPR = 6 gives a less

smooth appearing surface. The smaller NPPR is, the more localized the surface becomes (although NPPR = 4 will probably be the least value practicable, and some local interpolants will likely become planes due to the minimum of three interpolation points being reached). In line with this comment, very smooth surfaces with small gradients will probably be amenable to larger NPPR, while surfaces with large gradients may be best approximated by taking NPPR smaller, thus localizing the behavior.

Table 2 gives the results of a series of problems with various numbers of points and values of NPPR. The data points were chosen by a pseudorandom number generator in the square $[ -\frac{1}{18} , \frac{19}{18} ]^2$, and the approximation was evaluated on a 33x33 grid (of 1089 points total) on $[0,1]^2$. We observe that increasing NPPR increases execution time while decreasing the amount of storage needed. Preprocessing time should be about proportional to $N^2$, but apparently there is a strong linear component for small N. Preprocessing time should also be about proportional to NPPR, although this is not readily apparent. Evaluation time should be nearly independent of N for large N, and proportional to NPPR, which is approximately shown.

The automatic grid selection process works well when the data is fairly uniform in (x,y) and lies nearly in a square region. If the data is very irregular, or lies in an oblong rectangular area, it will probably be useful to explore results with a user specified grid to obtain better coverage by rectangles which are not too oblong. Limited experience has been accumulated in these situations.

6.0   How to Obtain this Program

A copy of this program, written in Fortran, and including a sample driver program, can be obtained from the author. To do so, send a (short) 1/2 inch tape to the author indicating the format desired.

14

## Acknowledgements

| x | y | f |
|------|-------|-----|
| 0.35 | 0.35 | 0.5 |
| -0.05 | 0.25 | 0.0 |
| 0.10 | -0.05 | 0.0 |
| 0.50 | 0.05 | 0.0 |
| 0.00 | 0.90 | 0.0 |
| 0.30 | 0.70 | 0.0 |
| 0.60 | 0.50 | 0.0 |
| 0.90 | 0.00 | 0.0 |
| 0.40 | 1.05 | 0.0 |
| 0.85 | 0.80 | 0.0 |
| 1.05 | 0.20 | 0.0 |
| 1.10 | 1.10 | 0.0 |

Table 1:   Data for "Cardinal" function.

| NPPR | N | $n_x = n_y$ | NWKU | NIWKU | Time (Preprocessing) | Time (Evaluation) |
|---|---|---|---|---|---|---|
| 6 | 60 | 5 | 334 | 273 | 4.2 | 14.3 |
| 6 | 100 | 7 | 619 | 506 | 7.5 | 17.6 |
| 6 | 500 | 18 | 3596 | 2893 | 70.5 | 21.2 |
| 6 | 1000 | 25 | 7441 | 6140 | 201.5 | 21.3 |
| 10 | 60 | 4 | 284 | 243 | 5.3 | 18.0 |
| 10 | 100 | 5 | 488 | 427 | 9.9 | 26.4 |
| 10 | 500 | 13 | 3014 | 2649 | 73.3 | 29.8 |
| 10 | 1000 | 19 | 6565 | 5804 | 202.6 | 31.7 |
| 15 | 60 | 3 | 224 | 199 | 6.8 | 23.0 |
| 15 | 100 | 4 | 414 | 373 | 12.4 | 32.8 |
| 15 | 500 | 11 | 2856 | 2591 | 91.6 | 39.3 |
| 15 | 1000 | 15 | 5920 | 5439 | 258.5 | 40.6 |

Table 2: Storage/Times for Various Size Problems.
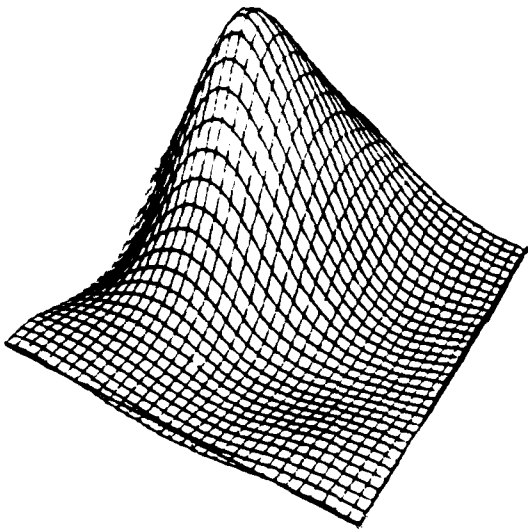
17

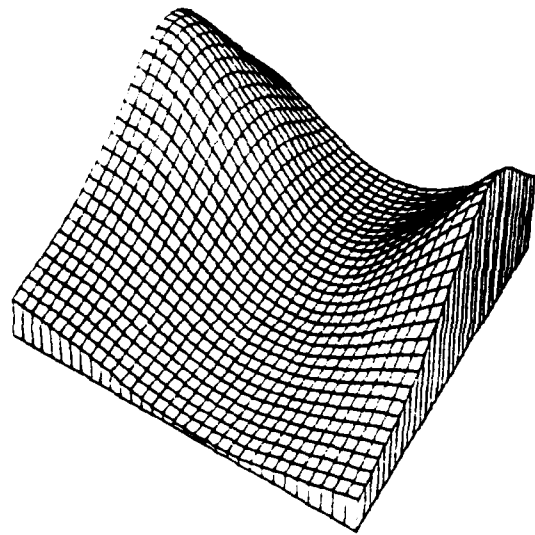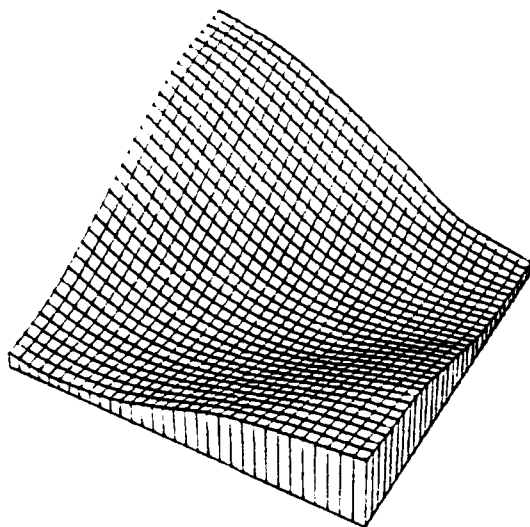Figure 1: Cardinal Function          Figure 2: Saddle Function

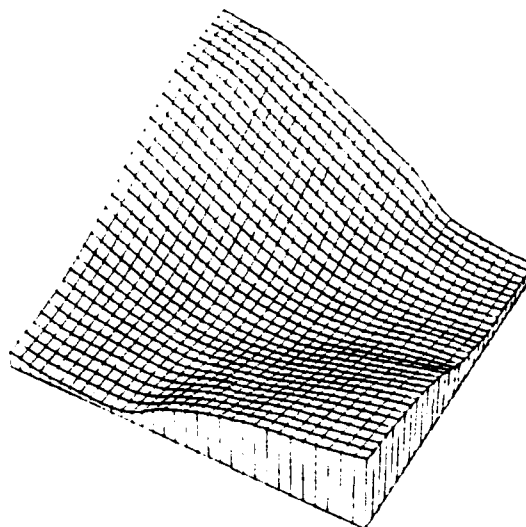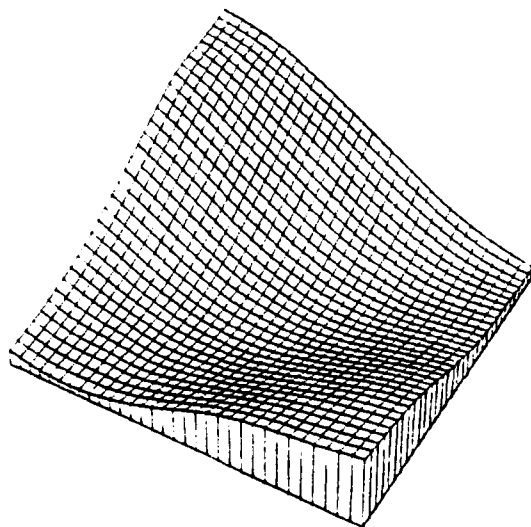Figure 3:   Parent Function
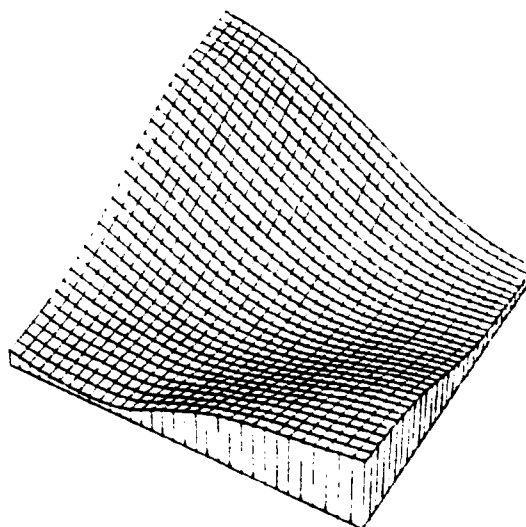


Figure 4:   NPPR = 6



Figure 5:   NPPR = 10



Figure 6:   NPPR = 15

19

# References

[1]  AKIMA, Hiroshi - "A Method of Bivariate Interpolation and Smooth
     Surface Fitting for Irregularly Distributed Data Points", ACM TOMS
     4 (1978) 148-159.

[2]  AKIMA, Hiroshi - "Algorithm 526: Bivariate Interpolation and Smooth
     Surface Fitting for Irregularly Distributed Data Points", ACM TOMS
     4 (1978) 160-164.

[3]  BARNHILL, R.E. - "Representation and Approximation of Surfaces",
     pp 69-120 in Mathematical Software III, J.R. Rice, ed., Academic Press,
     1977.

[4]  DUCHON, Jean - "Fonctions - Spline du Type Plaque Mince en Dimencion 2",
     Report #231, Univ. of Grenoble, 1975.

[5]  DUCHON, Jean - "Fonctions - Spline a Energie Invariate par Rotation",
     Report #27, Univ. of Grenoble, 1976.

[6]  DUCHON, Jean - "Interpolation des fonctions de deux Variables Suivant
     le Principe de la Flexion des Plaques Minces", R.A.I.R.O. Analyse
     Numerique 10 (1976) 5-12.

[7]  DUCHON, Jean - "Splines Minimizing Rotation - Invariant Semi-Norms
     in Sobolev Spaces", pp 35-100 in Constructive Theory of Functions of
     Several Variables, W. Schempp and K. Zeller, eds., Lecture Notes in
     Math 571, Springer, 1977.

[8]  FORSYTHE, G.E.; MALCOLM, M.A.; MOLER, C.B. - Computer Methods for Math-
     ematical Computations, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[9]  FRANKE, Richard - "Locally Determined Smooth Interpolation at Irregularly
     Spaced Points in Several Variables", JIMA 19 (1977) 471-482.

[10] FRANKE, Richard - "A Critical Comparison of Some Methods for Interpolation
     of Scattered Data", Naval Postgraduate School, TR#NPS-53-79-003, 1979
     (Available from NTIS, #AD-A081 688/4).

[11] FRANKE, Richard - "Scattered Data Interpolation: Tests of Some Methods",
     to appear in Math. Comp.

[12] FRANKE, Richard; NIELSON, Gregory - "Smooth Interpolation of Large Sets
     of Scattered Data", Int. J. Num. Meth. Engrg. 15 (1980) 1691-1704.

[13] HARDER, R.L.; DESMARAIS, R.N. - "Interpolation Using Surface Splines",
     J. Aircraft 9 (1972) 189-191.

[14] LAWSON, C.L. - "Software for $C^1$ Surface Interpolation", pp 159-192 in
     Software III, J.R. Rice, ed., Academic Press, 1977.

[15] LITTLE, Frank, University of Utah CAGD Report (forthcoming).

[16]   MEINGUET, Jean - "Multivariate Interpolation at Arbitrary Points Made
       Simple", ZAMP 30 (1979) 292-304.

[17]   MEINGUET, Jean - "An Intrinsic Approach to Multivariate Spline Interpol-
       ation at Arbitrary Points", pp 163-190 in Polynomial and Spline
       Approximation, B.N. Sahney, ed., D. Reidel Publishing Co., 1979.

[18]   NIELSON, Gregory M. - "A Method for Interpolating Scattered Data Based
       Upon a Minimum Norm Network", a manuscript.

[19]   SABIN, M.A. - "Contouring - A Review of Methods for Scattered Data",
       pp 63-85 in Mathematical Methods in Computer Graphics and Design,
       K.W. Brodlie, ed., Academic Press, 1980.

[20]   SCHUMAKER, L.L. - "Fitting Surfaces to Scattered Data", in Approximation
       Theory II, pp 203-268, G.G. Lorentz, C.K. Chui, L.L. Schumaker, eds.,
       Academic Press, 1976.

Appendix

```
C     SUBROUTINE LOTPS (MODE,NPPR,NPI,XI,YI,FI,NXO,XO,NYO,YO,IWK,NIWK,     LOT  1
C    1 NIWKU,WK,NWK,NWKU,FO,KER)                                          LOT  2
C                                                                         LOT  3
C     THIS SUBROUTINE SERVES AS A USER INTERFACE TO THE SET OF            LOT  4
C     SUBROUTINES THAT IMPLEMENT FRANKE'S METHOD OF SURFACE INTERPO-      LOT  5
C     LATION.  RECTANGLULAR REGIONS ARE USED WITH PRODUCT CUBIC           LOT  6
C     HERMITE WEIGHT FUNCTIONS.   THE RECTANGLES ARE CHOSEN IN AN ATTEMPT LOT  7
C     TO OBTAIN ABOUT NPPR POINTS IN EACH REGION.  THE SAME NUMBER OF     LOT  8
C     GRID LINES IS USED IN EACH DIRECTION.  LOCAL INTERPOLATION FUNC-    LOT  9
C     TIONS ARE THE THIN PLATE SPLINES DESCRIBED BY DUCHON AND OTHERS.    LOT 10
C     A DESCRIPTION OF THE METHOD AND REFERENCES APPEAR IN                LOT 11
C                                                                         LOT 12
C     SMOOTH INTERPOLATION OF SCATTERED DATA BY LOCAL THIN PLATE SPLINES  LOT 13
C     NAVAL POSTGRADUATE SCHOOL REPORT  NPS-53-81-002                     LOT 14
C                                                                         LOT 15
C         RICHARD FRANKE                                                  LOT 16
C         DEPARTMENT OF MATHEMATICS                                       LOT 17
C         NAVAL POSTGRADUATE SCHOOL                                       LOT 18
C         MONTEREY, CALIFORNIA  93940                                     LOT 19
C         (408)646-2758 / 2206                                            LOT 20
C                                                                         LOT 21
C     DIFFICULTIES AND SUCCESSES WITH THIS PROGRAM SHOULD BE              LOT 22
C     COMMUNICATED TO THE AUTHOR.                                         LOT 23
C                                                                         LOT 24
C     THE ARGUMENTS ARE AS FOLLOWS.                                       LOT 25
C                                                                         LOT 26
C     MODE - INPUT.   INDICATES THE STATUS OF THE CALCULATION.            LOT 27
C        = 1,  SET UP THE PROBLEM.  COMPUTE THE COEFFICIENTS              LOT 28
C              FOR THE LOCAL APPROXIMATIONS BY THIN PLATE                 LOT 29
C              SPLINES, AND RETURN THE GRID OF INTERPOLATED               LOT 30
C              FUNCTION VALUES INDICATED BY NXO, XO, NYO, YO              LOT 31
C              IN FO.                                                     LOT 32
C        = 2,  THE PROBLEM HAS BEEN SET UP PREVIOUSLY.  CALCU-            LOT 33
C              LATE THE GRID OF INTERPOLATED POINTS INDICATED             LOT 34
C              BY NXO, XO, NYO, YO IN FO.  THE PROGRAM ASSUMES            LOT 35
C              THAT THE ARRAYS XI, YI, IWK, AND WK ARE                    LOT 36
C              UNCHANGED FROM THE PREVIOUS CALL.
```

22

```
C  NPPR  - INPUT.   DESIRED AVERAGE NUMBER OF POINTS PER REGION.       LOT 37
C                   THE SUGGESTED VALUE IS TEN.  SHOULD BE AT LEAST    LOT 38
C                   FOUR.  VALUES LARGER THAN FIFTEEN COULD REQUIRE    LOT 39
C                   MINOR PROGRAM MODIFICATIONS TO ALLOW MORE          LOT 40
C                   STORAGE FOR THE EQUATION SOLVER.   THIS DEPENDS    LOT 41
C                   ON THE DISPOSITION OF THE POINTS.                  LOT 42
C                   IF THE USER WISHES TO SPECIFY HIS OWN GRID LINESLOT 43
C                   X TILDA AND Y TILDA, HE MAY DO SO BY SETTING       LOT 44
C                   NPPR = 0 AND SETTING NECESSARY VALUES IN THE       LOT 45
C                   ARRAYS IWK AND WK, AS NOTED BELOW.  DATA WHICH      LOT 46
C                   HAS A POOR DISTRIBUTION OVER THE REGION OF         LOT 47
C                   INTEREST SHOULD PROBABLY HAVE THE GRID SPECIFIEDLOT 48
C                   THIS IS ALSO ADVISABLE IF THE X-Y POINTS OCCUR     LOT 49
C                   ON LINES.                                          LOT 50
C  NPI   - INPUT.   NUMBER OF INPUT DATA POINTS.                       LOT 51
C  XI    - \                                                          LOT 52
C  YI    - INPUT.   THE DATA POINTS (XI,YI,FI), I=1,...,NPI.           LOT 53
C  FI    - /                                                          LOT 54
C  NXO   - INPUT.   THE NUMBER OF XO VALUES AT WHICH THE INTERP-       LOT 55
C                   OLATION FUNCTION IS TO BE CALCULATED.              LOT 56
C  XO    - INPUT.   THE VALUES OF X AT WHICH THE INTERPOLATION         LOT 57
C                   FUNCTION IS TO BE CALCULATED.  THESE SHOULD        LOT 58
C                   BE IN INCREASING ORDER FOR MOST EFFICIENT          LOT 59
C                   EVALUATION.                                        LOT 60
C  NYO   - INPUT.   THE NUMBER OF YO VALUES AT WHICH THE INTERP-       LOT 61
C                   OLATION FUNCTION IS TO BE CALCULATED.              LOT 62
C  YO    - INPUT.   THE VALUES OF Y AT WHICH THE INTERPOLATION         LOT 63
C                   FUNCTION IS TO BE CALCULATED.  THESE SHOULD        LOT 64
C                   BE IN INCREASING ORDER FOR MOST EFFICIENT          LOT 65
C                   EVALUATION.                                        LOT 66
C  IWK   - INPUT AND OUTPUT.  THIS ARRAY IS OUTPUT WHEN MODE = 1       LOT 67
C                   AND IS INPUT WHEN MODE = 2.   THIS MUST BE         LOT 68
C                   AN ARRAY DIMENSIONED APPROXIMATELY 6*NPI.          LOT 69
C                   WHEN NPPR IS INPUT AS ZERO THE USER MUST           LOT 70
C                   SPECIFY THE NUMBER OF VERTICAL GRID LINES (THE     LOT 71
C                   NUMBER OF X TILDA VALUES) IN IWK(1) AND THE        LOT 72
C                   NUMBER OF HORIZONTAL GRID LINES (THE NUMBER OF     LOT 73
C                   Y TILDA VALUES) IN IWK(2).                         LOT 74
```

23

```
C  NIWK - INPUT.   ON ENTRY WITH MODE = 1 THIS MUST BE SET TO THE      LOT 75
C                  DIMENSION OF THE ARRAY IWK.                         LOT 76
C  NIWKU- OUTPUT.  THE ACTUAL NUMBER OF LOCATIONS USED IN THE          LOT 77
C                  ARRAY IWK.                                          LOT 78
C  WK   - INPUT AND OUTPUT.  THIS ARRAY IS OUTPUT WHEN MODE = 1        LOT 79
C                  AND IS INPUT WHEN MODE = 2.  THIS MUST BE AN        LOT 80
C                  ARRAY DIMENSIONED APPROXIMATELY 7*NPI LOCATIONS.    LOT 81
C                  LOCATIONS.                                          LOT 82
C                  WHEN NPPR IS INPUT AS ZERO THE USER MUST SPECIFY    LOT 83
C                  THE VALUES OF X TILDA AND Y TILDA AS FOLLOWS.       LOT 84
C                  WK(2), ... , WK(NXG+1) ARE THE NXG (= IWK(1))       LOT 85
C                  X GRID VALUES, X(I) TILDA, IN INCREASING ORDER.     LOT 86
C                  TYPICALLY WK(1) = MIN X(I), ALTHOUGH IT NEED        LOT 87
C                  NOT BE.  WK(1) MUST BE LESS THAN OR EQUAL TO        LOT 88
C                  WK(2), AND SHOULD BE LESS THAN OR EQUAL TO          LOT 89
C                  MIN X(I).  WK(NXG+2) IS USUALLY MAX X(I), AL-       LOT 90
C                  THOUGH IT NEED NOT BE.  WK(NXG+2) MUST BE           LOT 91
C                  GREATER THAN WK(NXG+1), AND SHOULD BE GREATER       LOT 92
C                  THAN OR EQUAL TO MAX X(I).                          LOT 93
C                  THE VALUES OF WK(NXG+3), ... , WK(NXG+NYG+4)        LOT 94
C                  ARE THE Y GRID VALUES, Y(I) TILDA, AND MUST         LOT 95
C                  SATISFY DUAL CONDITIONS.                            LOT 96
C  NWK  - INPUT.   ON ENTRY WITH MODE = 1 THIS MUST BE SET TO THE      LOT 97
C                  DIMENSION OF THE ARRAY WK.                          LOT 98
C  NWKU - OUTPUT.  THE ACTUAL NUMBER OF LOCATIONS USED IN THE          LOT 99
C                  ARRAY WK.                                           LOT 100
C  FO   - OUTPUT.  VALUES OF THE INTERPOLATION FUNCTION AT THE         LOT 101
C                  GRID OF POINTS INDICATED BY NXO, XO, NYO, YO.       LOT 102
C                  FO IS ASSUMED TO BE DIMENSIONED (NXO,NYO) IN THE    LOT 103
C                  CALLING PROGRAM.                                    LOT 104
```

24

```
C     KER    - OUTPUT.    RETURN INDICATOR.                          LOT 105
C              = 0,       NORMAL RETURN.                             LOT 106
C              =-1,       PROBLEM HAS NOT BEEN PREVIOUSLY SET UP (LOTPS  LOT 107
C                         CALLED WITH MODE = 1)                      LOT 108
C              = 1,       ERROR RETURN FROM CLOTPS, SINGULAR MATRIX IN THELOT 109
C                         CALCULATION OF THE THIN PLATE SPLINES.     LOT 110
C              = 2,       ERROR RETURN FROM CLOTPS.  SOME RECTANGLE    (I,J)LOT 111
C                         HAS MORE THAN THE ALLOWED NUMBER OF POINTS  LOT 112
C                         ASSOCIATED WITH IT.  SEE CLOTPS FOR THE FIX. LOT 113
C              = 3,       PREVIOUS ERROR RETURN FROM CLOTPS HAS NOT BEEN LOT 114
C                         CORRECTED.                                  LOT 115
C              = 4,       IWK AND/OR WK ARRAYS HAVE NOT BEEN DIMENSIONED LOT 116
C                         LARGE ENOUGH IN THE CALLING PROGRAM.  REDIMEN- LOT 117
C                         SION IWK AND WK TO AT LEAST THE SIZE INDICATED LOT 118
C                         BY NIWKU AND NWKU, RESPECTIVELY.            LOT 119
C              = 5,       MODE IS OUT OF RANGE.                       LOT 120
C                                                                     LOT 121
C     SUBPROGRAMS CALLED BY THIS ROUTINE ARE    GRID, LOCLIP, CLOTPS, LOT 122
C     AND EVLTPS.  ALSO REQUIRED ARE  USRTA  FROM THE IMSL LIBRARY AND LOT 123
C     THE ROUTINES DECOMP AND SOLVE  FROM FORSYTHE, MALCOLM, AND MOLER.LOT 124
```

25

DISTRIBUTION LIST

| | |
|---|---|
| Defense Technical Information Center<br>Cameron Station<br>Alexandria, VA 22314 | 2 |
| Dudley Knox Library<br>Naval Postgraduate School<br>Monterey, CA 93940 | 2 |
| Dean of Research<br>Naval Postgraduate School<br>Monterey, CA 93940 | 2 |

Department of Mathematics

| | |
|---|---|
| C. O. Wilde, Chairman | 1 |
| Professor F. D. Faulkner | 1 |
| Professor A. Schoenstadt | 1 |
| Professor R. Franke | 10 |

Naval Postgraduate School
Monterey, CA 93940

| | |
|---|---|
| Dr. Richard Lau<br>Office of Naval Research<br>1030 East Green St.<br>Pasadena, CA 91106 | 1 |
| Professor R. E. Barnhill<br>Department of Mathematics<br>University of Utah<br>Salt Lake City, UT 84112 | 1 |
| Professor G. M. Nielson<br>Department of Mathematics<br>Arizona State University<br>Tempe, AZ 85281 | 1 |
| Chief of Naval Research<br>ATT: Mathematics Program<br>ARlington, VA 22217 | 2 |
| Rosemary E. Chang<br>Sandia Laboratories<br>Applied Mathematics, Division 2-8235<br>Livermore, CA 94550 | 1 |
| L. H. Seitelman<br>Pratt and Whitney Aircraft<br>400 Main Street<br>East Hartford, CT 06108 | 1 |
| William W. Bozek<br>Engineering Data Processing<br>Philips ECG, Inc.<br>Johnston Street<br>Seneca Falls, NY 13148 | 1 |

William J. Gordon                                          1
Department of Mathematical Sciences
Drexel University
Philadelphia, PA 19104

Loren N. Argabright                                        1
Department of Mathematical Sciences
Drexel University
Philadelphia, PA   19104

# END

## DATE
## FILMED

# 5-81

# DTIC